7Ч

УДК 004.4'42

Ю.В. Чернухин, М.Ю. Поленов, Д.В. Булгаков
Технологический институт Южного федерального университета в г. Таганроге, Россия
chernukhin@dce.tsure.ru, polenov@dce.tsure.ru, buda@mail.ru

Об одном подходе к интеллектуализации средств мультиязыковой трансляции при решении задач САПР

Рассматривается подход к интеллектуализации разработанной ранее среды многоязыковой трансляции (мультитранслятора) посредством подключения к ней экспертной системы для поддержки автоматического перевода моделей устройств на различных уровнях проектирования систем на кристалле. Реализация данного подхода позволяет расширить функциональные возможности мультитранслятора и САПР СБИС, использующих его в качестве средства перевода и синтеза описаний моделей.

Введение

При разработке систем на кристалле (СНК) современные САПР СБИС позволяют реализовать полный цикл синтеза моделей устройств посредством использования собственных компонент-синтезаторов, генерирующих описания, реализованные на одном из языков описания аппаратуры (ЯОА), на каждом этапе проектирования [1], [2]. Данные САПР традиционно ориентированы на определенные версии ЯОА, и добавление поддержки нового или изменение версии существующего языка возможно только путем модернизации самой системы проектирования и становится доступным пользователям только при выходе ее новой версии.

Для обеспечения возможности оперативного расширения набора ЯОА для современных САПР предложено использовать разработанные средства многоязыковой трансляции на различных этапах проектирования СБИС [3].

Подход, используемый в средствах многоязыковой трансляции, названных мультитранслятором (МТ) [4], позволяет отделить процесс описания правил трансляции от технических вопросов ее организации. Процесс создания нового транслятора в МТ сводится к описанию правил трансляции для каждой пары языков в трансляционном модуле (ТМ), подключаемом к инвариантному относительно решаемой задачи ядру мультитранслятора. Данный подход позволяет успешно применять МТ не только для решения задач перевода моделей систем моделирования [5] и разработки компиляторов с языков высокого уровня [6], но и использовать его при проектировании СНК, например, для конверсии алгоритмических описаний на высокоуровневых языках описания аппаратуры (таких, как SystemC и VHDL) [3] или для RTL-синтеза поведенческого описания.

Применение мультитранслятора для перевода моделей на различных этапах проектирования позволяет сделать САПР СБИС более гибкой и универсальной. Но следует подчеркнуть, что в процессе решения некоторых подзадач перевода возникают проблемы, которые невозможно решить стандартными средствами МТ. Такие проблемы, например, возникают в случаях, когда мультитранслятору не достаточно входных данных для продолжения процесса мультитрансляции или когда подзадача

перевода имеет более одного допустимого решения. Выходом из данной ситуации является привлечение пользователя, разрешающего неоднозначность перевода при помощи диалогового *Мастера трансляции*, либо применение дополнительных интеллектуальных средств МТ. В качестве такого средства интеллектуализации мультитранслятора предлагается использовать экспертную систему (ЭС) [7], которая способна принять решение вместо специалиста в области проектирования СБИС, причем решение ЭС не должно уступать по качеству и эффективности решению такого специалиста.

1 Применение экспертных систем в задачах конверсии моделей на языках описания аппаратуры

Процесс мультитрансляции представляет собой генерацию для каждой конструкции или набора конструкций входного языка некоторой конструкции или набора конструкций выходного языка по правилам, описанным в трансляционном модуле МТ. Генерация выходного кода производится в процессе построения синтаксического дерева входного языка. Если параметры разбираемой конструкции входного языка следуют в порядке, отличном от аналогичных параметров конструкции выходного языка, то генерация выходного кода не может быть осуществлена последовательно по мере разбора синтаксических символов входного языка. В таком случае возможно применение либо отложенного формирования генерируемого кода, либо вставка кода посредством текстовых меток [3].

Для передачи данных между грамматическими правилами ТМ, описывающими разбор конструкций входного языка, целесообразно использовать стек, который может хранить как простые строковые значения, так и более сложные структуры данных. При этом разрешается передавать посредством стека между символами синтаксического дерева параметры различной сложности (отдельные строки, структуры данных, ссылки на таблицы и т.п.). Это позволяет снизить потребность в ресурсах инструментальной рабочей станции, поскольку такие структуры данных создаются только при разборе тех символов синтаксического дерева, где они действительно необходимы, и высвобождаются по мере выхода из соответствующих символов. Для работы со стеками и построения динамических массивов и таблиц любой степени сложности мультитранслятор позволяет использовать модули расширения и динамические подключаемые библиотеки с необходимыми функциями, реализация которых может быть осуществлена пользователем на различных языках программирования высокого уровня.

Рассмотренные подходы позволяют выполнять мультитрансляцию различных конструкций входного языка описания аппаратуры (например, SystemC) в конструкции выходного языка (VHDL) [3]. При этом следует учесть особенности отдельных конструкций ЯОА, как со стороны их синтаксического способа представления, так и со стороны их применимости в различных частях программного кода. Также должны быть определены ограничения для синтаксиса входных конструкций. Это связано с тем, что не все конструкции, поддерживаемые входным языком, могут быть переведены в соответствующие конструкции выходного языка.

В общем случае набор транслируемых конструкций ЯОА можно разделить на группы (программные модули устройств, структурные описания, операторы, выражения), алгоритмы мультитрансляции которых могут быть реализованы на основании знаний об особенностях построения аналогичных конструкций входного и выходного языков [3].

Структурное описание в ЯОА предназначено для объединения отдельных экземпляров устройств (компонент) в единую схему. Например, в языке SystemC [8] отдельные компоненты декларируются в разделе объявлений программного модуля устройства. Синтаксис их объявления полностью совпадает с объявлением других объектов модуля [9], за исключением того, что имя типа компоненты может быть произвольным. При этом необходимо учитывать, что для генерации соответствующей компоненты выходного языка трансляционному модулю мультитранслятора необходима информация о количестве и типах портов такой компоненты, а также ТМ должен иметь информацию об используемых внешних компонентах.

В ситуации, когда в процессе разбора входного языка было распознано имя некоторой компоненты, можно предложить следующие варианты работы мультитранслятора:

1) МТ формирует имя и набор параметров (интерфейс) компоненты выходного языка, опираясь на описание компоненты в таблице компонент пакета транслируемых файлов. Такая таблица формируется в процессе трансляции подключенных файлов с описанием исходного кода самих компонент. Таким образом, до трансляции объявления компоненты в текущем программном коде устройства мультитранслятор должен предварительно транслировать код самой компоненты и иметь необходимые данные о количестве и типе ее портов;

2) мультитранслятор имеет описание компоненты в библиотеке соответствия компонент для пары языков, например, для SystemC и VHDL. Поскольку подключаемые компоненты на SystemC могут являться собственностью сторонних разработчиков и распространяться без исходных кодов, то в таком случае мультитранслятор должен иметь специальную библиотеку, в которой были бы заранее размещены описания интерфейсов компонент на SystemC и описания интерфейсов аналогичных устройств, реализованных сторонними разработчиками на VHDL. В процессе трансляции МТ производит замену имен компонент входного языка описания аппаратуры соответствующими именами выходного языка и использует определенный порядок параметров компонент, опираясь на данные из библиотеки соответствия. Недостаток такого подхода заключается в необходимости составления библиотек соответствия, которые для данной задачи может сформировать только разработчик, знающий все используемые в транслируемом проекте сторонние компоненты SystemC и их аналоги для VHDL;

3) мультитранслятор при помощи модуля-мастера запрашивает у пользователя в диалоговом режиме имя и интерфейс аналога для найденной компоненты входного языка. На рис. 1 показано условное графическое изображение такого диалогового мастера на примере языков SystemC и VHDL. На этом рисунке символами вопросов обозначены поля ввода соответствующих параметров для компоненты выходного языка. Пользователь должен ввести имя и библиотеку некоторой компоненты на VHDL, которая является функциональным аналогом найденной компоненте на языке SystemC, а также задать соответствие портов компонент. Область применения мастеров не ограничивается установлением соответствия компонент. Они могут быть использованы для разрешения различных неоднозначностей в процессе трансляции программного кода устройств. Основной недостаток данного подхода – необходимость участия человека-эксперта в процессе мультитрансляции, что делает недоступным применение МТ пользователями, не имеющими углубленных знаний в области языков описания аппаратуры.

Другой специфичной подзадачей в области конверсии моделей на ЯОА, является мультитрансляция типов данных, сигналов и портов устройств. Часто возникают ситуации, когда множество значений аналогичных типов данных для входного и выходного

языков совпадают не полностью. Так множество значений выходного типа данных может являться подмножеством входного типа. Таким образом, в процессе мультитрансляции выражений должна происходить замена входных константных значений, аналоги которых не существуют во множестве значений соответствующего типа данных выходного языка, некоторыми другими допустимыми значениями. В табл. 1 показано соответствие входных и выходных значений типов данных стандартной логики для языков SystemC и VHDL.

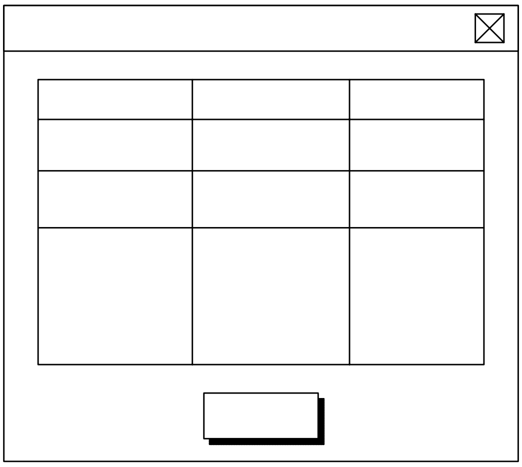


Рисунок 1 – Диалоговое окно мастера определения аналога компоненты

Таблица 1

Допустимые значения std_logic (входной код VHDL)	Допустимые значения sc_logic (выходной код SystemC)	Описание
‘0’	‘0’	Логический ноль
‘1’	‘1’	Логическая единица
‘X’	‘X’	Неизвестное значение
‘Z’	‘Z’	Z-состояние
‘U’ , ‘W’, ‘L’, ‘H’, ‘-’	Не поддерживаются	Другие значения

Как видно из табл. 1, множество значений выходного типа данных является сокращенным по отношению к множеству входных значений. Следовательно, в процессе мультитрансляции необходимо каждое неподдерживаемое выходным языком значение заменять разрешенным. Для полной автоматизации такого процесса предлагается применение подключаемой экспертной системы, которая позволит определять значения констант автоматически, оперируя знаниями о способах подстановки значений констант в зависимости от контекста их применения в программном коде модели устройства.

Подход на основе экспертной системы также может использоваться при приведении типов для операндов выражений.

Например, язык SystemC позволяет выполнять операции над объектами различных типов с использованием неявного приведения типов [9], в отличие от выражений на VHDL, где все типы операндов и допустимые для них операции заданы в соответствующих подключаемых библиотеках. Следовательно, не все входные выражения

на SystemC могут быть транслированы в соответствующие выходные выражения на VHDL. Эту проблему можно решить при помощи дополнительных операторов, поддерживаемых языком VHDL для явного преобразования типов объектов. При этом для трансляции выражений могут быть применены различные наборы операторов приведения типов данных.

Процесс обработки выражений и решения задачи доопределения операндов может быть реализован мультитранслятором с помощью диалоговых мастеров, позволяющих выбрать для каждого операнда необходимого оператора приведения типа. Но более эффективным представляется решение данной задачи при помощи специализированной экспертной системы, позволяющей выполнить приведение типов автоматически, основываясь на знаниях специалистов-экспертов в области разработки СБИС и с учетом различных критериев, таких как минимизация числа операторов приведения в выражении или применения операторов, требующих минимальных затрат ресурсов проектируемого устройства при реализации СБИС. Экспертная система должна принимать решения при возникновении ситуаций неопределенности перевода и должна иметь возможность пополнять собственную базу знаний [7].

На рис. 2 изображена организация подключения и взаимодействия мультитранслятора с экспертной системой.

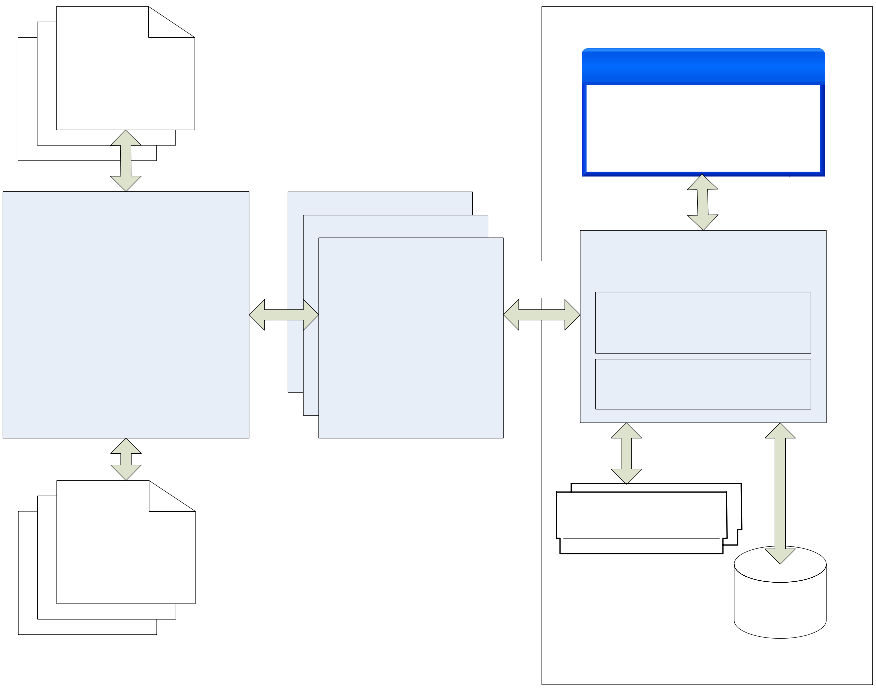


Рисунок 2 – Структура взаимодействия мультитранслятора и экспертной системы

По окончании разбора выражения в ТМ мультитранслятора в ЭС посредством прикладного интерфейса (API) передается разобранное выражение в специальном виде с указанием типов операндов.

Далее работу по обработке выражения берет на себя сама ЭС. База знаний экспертной системы при этом должна содержать набор правил разбора входного выражения, размещаемый в рабочей памяти. Правила записываются на специальном языке символьного программирования и имеют вид: *Если <Условие> тогда <Действия>*. Каждое правило может иметь некоторую оценку, назначаемую при

вводе правил или формируемую автоматически в процессе обучения системы. Правила поочередно применяются при помощи механизма логического вывода ЭС к входному выражению, при этом если условия правила применимы, то выполняется некоторое действие по модификации выражения в рабочей памяти. Условия правил могут относиться как к конкретным, рядом расположенным операндам выражения, связанным конкретной операцией, так и касаться статистических данных обо всем выражении (например, количества наиболее часто повторяющихся типов данных операндов).

После применения каждого правила к выражению, оно может быть модифицировано различными способами:

– два объекта, имеющие исходный тип данных и связанные операцией *X*, могут быть заменены новым объектом *R* типа *TR*, если язык VHDL поддерживает операцию *X*, результат которой имеет тип *TR*;

– объект *A* типа *T0* может быть приведен к типу *TR* оператором приведения типов *P* и заменен объектом *R* типа *TR*, если язык VHDL поддерживает оператор *P* приведения типа *T0* к типу *TR*.

В результате применения последовательности правил к входному выражению в рабочей памяти ЭС останется единственный операнд определенного типа данных. Если тип этого операнда совпадает с требуемым типом выражения (передается в ЭС при передаче выражения для обработки), то задача разбора выражения считается завершенной и ЭС передает трансляционному модулю МТ результат работы в виде специальной строки, формируемой в процессе применения правил. В противном случае производится откат на одну операцию модификации выражения и применяется другое, удовлетворяющее его условиям правило, если последнее существует. Если же правило не существует, то снова производится откат, и так до тех пор, пока не будет найдено новое правило. Фактически данная система является продукционной, состоящей из множества правил (продукционная память) и интерпретатора правил, позволяющей производить эвристический вывод выражений с возможностью обучения.

Правила формируются в процессе обучения ЭС пользователем-экспертом. При обучении на ее вход подается выборка из различных выражений, включающих в себя разнообразные комбинации типов данных операндов. При разборе выражения в процессе обучения эксперт посредством подсистемы диалога (интеллектуального редактора) [10] добавляет правила логического вывода, которые он формирует, опираясь на свои знания о процессе приведения типов данных. Редактор должен предоставлять эксперту развитые средства для пополнения базы знаний ЭС, при этом тонкости реализации системы должны быть скрыты от эксперта.

Например, эксперт может знать, что в выходном языке описания аппаратуры существует операция сложения операнда типа вектор бит и константы единицы, и при этом результат такого выражения будет вектором бит. В таком случае эксперт вносит правило «*Если операнд типа вектор бит и операнд, равный константе единицы, объединены операцией сложения, тогда заменить это подвыражение операндом типа вектор бит*».

Каждому конкретному состоянию выражения в рабочей памяти ЭС могут по условию подходить одновременно несколько правил. Для разрешения таких конфликтов в процессе разбора выражения ядро ЭС использует некоторые эвристические оценки, позволяющие ЭС выбирать одно из наиболее подходящих правил. Оценка может формироваться различными способами. Одним из таких способов может быть присвоение каждому правилу числа, характеризующего ресурсоемкость оператора

приведения типов, применяемого к операнду в выражении. Каждый оператор приведения типов ЯОА при реализации на СБИС будет заменен схемой, занимающей некоторый объем ресурсов. Объем ресурсов может быть представлен численным значением, которое вводится экспертом для каждого нового правила в процессе обучения ЭС. Другой оценкой может служить относительный приоритет каждого правила. Так, при возникновении конфликта в процессе обучения ЭС эксперт может выбрать из списка правил, подходящих по условию для данной ситуации, одно правило, которое он считает наиболее подходящим, при этом рейтинг этого правила относительно других, предложенных эксперту, автоматически наращивается.

В процессе автоматического разбора выражений экспертная система будет сама выходить из конфликтов, выбирая наиболее приоритетные и имеющие наименьшую ресурсоемкость правила. Кроме приведенных типов оценок, могут быть использованы и другие, например оценка общего количества операторов приведения, разрядность операндов, к которым применяется операция приведения типов.

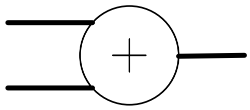
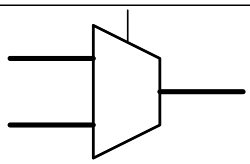
2 Интеллектуализация мультитрансляции в задачах RTL-синтеза моделей на языках описания аппаратуры

Предложенный подход к интеллектуализации средств мультитрансляции может быть использован как при конверсии ЯОА-программ, представленных на архитектурном уровне абстракции, так и для перевода моделей на более низких уровнях, например, для синтеза поведенческих моделей цифровых устройств на уровне регистровых передач.

Синтез уровня регистровых передач (RTL) представляет собой процесс трансляции программы алгоритмического уровня, заданной набором конструкций определенного языка описания аппаратуры, в набор соответствующих укрупненных примитивов из выбранного базиса. При этом совокупность примитивов описывает функциональную схему, которая далее может быть использована другими средствами синтеза для последующих этапов проектирования системы на кристалле (например, для логического и физического синтеза) и реализована в СБИС [1]. Примитивы RTL-представления должны быть описаны специальным RTL-языком, например, может быть использован один из известных форматов обмена проектами электронных схем (EDIF, CIF, DXF и др.).

В качестве языка, генерируемого ТМ мультитранслятора, был предложен упрощенный язык, примеры конструкций которого, с соответствующими им функциональными элементами, приведены в табл. 2.

Таблица 2 – Примеры конструкций RTL-языка и соответствующих функциональных элементов

Модель	Текстовое RTL-обозначение	Функциональные элементы
N-разрядный знаковый сумматор	`ADD( in A:<Wire>,` `in B:<Wire>,` `out Q:<Wire> )`	
Двухвходовый N-разрядный мультиплексор	`MUX( in A:<Wire>,` `in B:<Wire>,` `in S:<Wire>,` `out Q:<Wire> )`	

Проведенные исследования по трансляции при помощи МТ набора высокоуровневых синтезируемых конструкций языка VHDL (выражений, операторов условия, операторов присваивания, операторов цикла и других) продемонстрировали необходимость использования ЭС для разрешения ситуаций неопределенности, связанных с построением реализуемых на СБИС логических схем.

Экспертные системы могут потребоваться для решения проблем реализации и оптимизации СБИС с учетом заданных ограничений (например, распараллеливание и синхронизация схем в зависимости от временных ограничений, задаваемых разработчиком устройств). Так, например, циклы в языке VHDL могут транслироваться: в повторяющиеся комбинационные схемы, количество которых совпадает с числом итераций цикла, или в синхронную схему, которая будет производить одинаковые вычисления последовательно на каждой итерации цикла для различных входных параметров [11]. Понятно, что выбор способа трансляции VHDL-кода зависит от характеристик СБИС выбранной для реализации схемы.

Одной из важнейших задач, которая может быть решена мультитранслятором при помощи ЭС, является процесс трансляции из низкоуровневого RTL-описания в высокоуровневое, например, на языке VHDL. При решении такой задачи возникает неопределенность при выборе способа трансляции тех или иных групп инструкций низкоуровневого описания схемы в алгоритмические конструкции ЯОА высокого уровня.

Например, последовательность операторов RTL-языка:

```
LATCH(in A: Wire_A[0], in E: Wire_B[0], out Q:Wire_C[0]);
LATCH(in A: Wire_A[1], in E: Wire_B[1], out Q:Wire_C[1]);
```

где LATCH – защелка с входом данных *A*, входом разрешения записи *E* и выходом *Q*, в терминах языка VHDL может быть описана различным образом:

1) как оператор условия

```
if B="11" then C<=A; end if;
```

2) как оператор цикла с условием

```
for i in 0 to 1 loop
    if B(i)='1' then C(i)<=A(i); end if;
end loop;
```

В данном случае задачей экспертной системы является интеллектуальное выделение подгрупп низкоуровневых инструкций и их объединение в высокоуровневые конструкции с оптимизацией кода по таким критериям, как общий объем кода, простота кода и другие.

Для создания такой ЭС возможно применение специальных инструментальных средств разработки экспертных систем и языков представления знаний (например, среды CLIPS [12], языков LISP и PROLOG).

Среда CLIPS предлагает эвристические и процедурные подходы для представления знаний и вывода новых фактов, используя для этого специальный LISP-подобный язык представления порождающих правил и язык описания процедур. Среда CLIPS позволяет взаимодействовать с внешними модулями и библиотеками, написанными на других языках программирования [12], что позволяет реализовать ее интеграцию с мультитранслятором. CLIPS имеет дело с правилами и фактами, описанными особым образом. В таком случае для организации взаимодействия и передачи данных в CLIPS

трансляционный модуль должен содержать дополнительные процедуры для формирования данных в формате, понятном системе CLIPS. Так, описанные ранее в качестве примера инструкции RTL-языка могут быть представлены в виде фактов CLIPS:

(**Instruction** (**Name** LATCH)(**InpA** Wire_A0)(**InpE** Wire_B0)(**OutQ**:Wire_C0))
(**Instruction** (**Name** LATCH)(**InpA** Wire_A1)(**InpE** Wire_B1)(**OutQ**:Wire_C1))

Среда CLIPS имеет средства для гибкого описания правил и разрешения конфликтов [12] при выборе очередного правила. Таким образом, в процессе разбора выражения факты, соответствующие низкоуровневым конструкциям, должны заменяться фактами, описывающими высокоуровневые конструкции ЯОА. Так, задача по разбору выражения считается завершенной, если в рабочей памяти CLIPS не останется ни одного факта низкоуровневой конструкции. Полученные в результате замен факты должны быть возвращены мультитранслятору, который, в свою очередь, должен преобразовать их к виду, соответствующему VHDL-коду. Иными словами, построенная на базе CLIPS экспертная система может служить эффективным средством интеллектуализации средств мультитрансляции в случае решения задач RTL-синтеза моделей на языках описания аппаратуры при проектировании систем на кристалле методами САПР.

Заключение

Предлагаемый подход к интеллектуализации мультитранслятора посредством подключения к нему экспертной системы расширяет его функциональные возможности, повышает универсальность, что чрезвычайно важно как при решении задач САПР СБИС, так и при решении других задач многоязыковой трансляции (перевод моделей систем моделирования, создание компиляторов с различных языков и т.п.).

В данном случае целесообразно применить единую экспертную систему взамен отдельных узкоспециализированных подсистем для различных задач мультиязыковой трансляции. Такая ЭС может быть реализована в виде подключаемого интерпретатора, использующего универсальный язык символьного или логического программирования. В зависимости от конкретной задачи трансляции МТ сможет подключать одну из существующих баз знаний и выполнять необходимые преобразования кодов посредством ЭС, подавая на вход данные в формате, понятном интерпретатору ЭС.

Дополнительно отметим, что рассмотренное в данной статье применение экспертных систем является одним из возможных подходов к интеллектуализации средств мультиязыковой трансляции. Для решения более сложных задач трансляции не исключена возможность применения и других подходов, связанных с такими направлениями искусственного интеллекта, как нейросетевые и эволюционные вычисления.

Литература

1. Максфилд К. Проектирование на ПЛИС. Архитектура, средства и методы / Максфилд К. – М. : Издательский дом «Додэка-XXI», 2007. – 408 с.
2. Томас Ф. САПР микроэлектроники. Этапы большого пути / Ф. Томас, А. Иванов // Электроника. Наука, Технология, Бизнес. – 2006. – № 3. – С. 82-85.
3. Чернухин Ю.В. Использование мультиязыковой трансляции при конверсии моделей, представленных на языках описания аппаратуры / Ю.В. Чернухин, М.Ю. Поленов, Д.В. Булгаков // Искусственный интеллект. – 2009. – № 4. – С. 462-471.

4. Чернухин Ю.В. Многоязыковая трансляция средств виртуального моделирования / Ю.В. Чернухин, В.Ф. Гузик, М.Ю. Поленов. – Ростов н/Д. : Изд-во ЮНЦ РАН, 2009. – 368 с.
5. Чернухин Ю.В. Подход к формированию внешних библиотек сред виртуального моделирования на базе мультиязыковой трансляции / Ю.В. Чернухин, В.Ф. Гузик, М.Ю. Поленов // Вестник компьютерных и информационных технологий. – 2008. – № 10. – С. 2-12.
6. Чернухин Ю.В. Возможности применения среды мультитрансляции в качестве компилятора компиляторов / Ю.В. Чернухин, М.Ю. Поленов, Д.В. Левченко // Искусственный интеллект. – 2008. – № 4. – С. 712-720.
7. Коротко и понятно об экспертных системах. Портал искусственного интеллекта, 2010. [Электронный ресурс]. – Режим доступа : http://www.aiportal.ru/articles/expert-systems/short-about-expsys.html.
8. SystemC. Version 2.0. User's Guide. Open SystemC Initiative (OSCI) [Электронный ресурс]. – Режим доступа : http://www.systemc.org
9. Describing Synthesizable RTL in SystemC. Open SystemC Initiative (OSCI) [Электронный ресурс]. – Режим доступа : http://www.systemc.org
10. Технология разработки экспертных систем [Электронный ресурс]. – Режим доступа : http://www.lnup.ru/devtechnology/
11. Sharp R. Higher-Level Hardware Synthesis / R. Sharp. – Berlin : Springer, 2004. – 198 p.
12. Капустина Е.А. Применение средств языка CLIPS для организации и представления знаний в экспертных системах [Электронный ресурс] / Е.А. Капустина. – Режим доступа : http://www.masters.donntu.edu.ua/2005/kita/kapustina/ind/clips.htm.

Ю.В. Чернухин, М.Ю. Полєнов, Д.В. Булгаков
Про один підхід до інтелектуалізації засобів мультимовної трансляції при розв'язанні задач САПР
Розглянуто підхід до інтелектуалізації розробленого раніше середовища багатомовної трансляції (мультитранслятора) через підключення до неї експертної системи для підтримки автоматичного перекладу моделей приладів на різних рівнях проектування систем на кристалі. Реалізація даного підходу дозволяє розширити функціональні можливості мультитранслятора і САПР НВІС, що використовують його як засіб перекладу та синтезу описів моделей.

Yu.V. Chernukhin, M.Yu. Polenov, D.V. Bulgakov
About One Approach to Intellectualization of Multilanguage Translation Tools for the EDA's Tasks Decision
The approach to intellectualization of previously developed the multilanguage translation environment (Multitranslator) by means of the expert system usage to support of automatic translation of devices' models at various levels of the System-On-A-Chip design is considered. Implementation of the given approach allows to expand functionalities of the Multitranslator and the VLSI EDA systems, using Multitranslator as a translation and synthesis tools of models' descriptions.

Статья поступила в редакцию 30.06.2010.